

Cómo utilizar secuencias de comandos de resolución de problemas de FCS y CRC para ACI

Contenido

[Introducción](#)

[Prerrequisitos Para ejecutar el script manualmente](#)

[Prerrequisitos para ejecutar el script desde el contenedor](#)

[Pasos para ejecutar los scripts](#)

Introducción

ACI sigue el Switching de Corte, lo que significa que el Paquete ya se reenvía antes de que se pueda calcular el CRC. Estos paquetes se suelen pisotear y reenviar como errores de salida. Debido a que ACI no descarta estos paquetes, el mismo paquete atraviesa el paquete y los contadores de CRC de estómago se incrementan en el trayecto. Esto no significa que todas las interfaces que ven el CRC sean defectuosas. Por lo tanto, es necesario realizar el triage adecuado para aislar el problemático puerto/SFP/fibra. El proceso de triage ahora automatizado a través de scripts Python, lo que facilita la resolución de problemas y evita las tareas manuales. El alcance de este documento es explicar cómo utilizar los scripts de automatización (véase adjunto).

Prerrequisitos Para ejecutar el script manualmente

La máquina cliente desde la que se ejecutará la secuencia de comandos debe cumplir los siguientes requisitos

- a. Python3 debe instalarse
- b. Acceso de red al dominio ACI
- c. ACI_CRC_required.txt (adjunto) a instalar. Este archivo se encuentra [aquí](#).

Descargue el archivo (ACI_CRC_required.txt) en el equipo cliente

Abra Terminal y ejecute el comando - pip3 install -r ACI_CRC_require.txt

```
ABCD-M-G24X:downloads abcd$ pip3 install -r ACI_CRC_requirements.txt
```

```
Collecting bcrypt==3.2.0 (from -r ACI_CRC_requirements.txt (line 1))
```

```
Downloading
```

```
https://files.pythonhosted.org/packages/bf/6a/0afb1e04aebd4c3ceae630a87a55fbfbbd94dea4eaf01e53d36743c85f02/bcrypt-3.2.0-cp36-abi3-macosx\_10\_9\_x86\_64.whl
```

```
Collecting cffi==1.14.6 (from -r ACI_CRC_requirements.txt (line 2))
```

```
Downloading
```

```
https://files.pythonhosted.org/packages/ca/e1/015e2ae23230d9de8597e9ad8c0b81d5ac181f08f2e6e75774b7f5301677/cffi-1.14.6-cp38-cp38-macosx\_10\_9\_x86\_64.whl (176kB)
```

```
|| 184kB 1.4MB/s
```

```
**snip**
```


En este momento, la secuencia de comandos 1 comienza a recopilar errores FCS/CRC del fabric cada cinco minutos (hasta el tiempo de finalización especificado anteriormente por el usuario) y guarda los datos en los archivos en la ruta especificada en la entrada anterior.

```
Enter the End Time until which the script runs(in the format of yyyy-mm-dd hh:mm, current time:2021-09-27 11:27.... maximum upto 2021-10-04 11:27): 2021-09-27 11:32 <<<<<
```

```
The script is executing .....
```

```
The script is executing .....
```

```
ABCD-M-G24X:downloads abcd$
```

4. Una vez ejecutada correctamente la primera secuencia de comandos, almacenará los archivos de datos sin procesar en la ubicación especificada por el usuario en el paso 2.

Verifique lo mismo que se muestra en el siguiente ejemplo.

```
ABCD-M-G24X:FCS_Checker kbosu$ pwd
/Users/abcd/Downloads/FCS_Checker
```

```
ls -l
total 16
-rw-r--r--@ 1 kbosu staff 1419 Sep 27 11:28 CRC_FCS_20210927_1128.txt
-rw-r--r--@ 1 kbosu staff 1419 Sep 27 11:33 CRC_FCS_20210927_1133.txt
ABCD-M-G24X:FCS_Checker abcd$
```

5. Ahora es el momento de ejecutar el segundo script (ACI_CRC_Parser.py) .

Script-2 va a utilizar los archivos creados por script-1 y trabajar más a fondo.

Introduzca la dirección IP OOB de uno de los APIC en un clúster determinado y sus credenciales.

Además, introduzca la misma ubicación de archivo que ha especificado en el paso 2 mientras ejecutaba el primer script.

```
ABCD-M-G24X:downloads abcd$ python3 ACI_CRC_Parser.py
```

```
Enter the IP address or DNS Name of APIC: 10.197.204.184
```

```
Enter the username: admin
```

```
Enter the password: *****
```

```
Trying to connect to APIC
```

```
Connection established to the APIC
```

```
Please enter the folder where files are stored
```

```
Please make sure we have at least two files exists in the directory where you have saved data
```

```
Enter the absolute path of the folder where the files are stored:/Users/abcd/Downloads/FCS_Checker/
```

```
You have CRC and FCS for the below date range
```

```
1.2021-09-27
```

```
Fetching first and last file of the same date 20210927
```

```
CRC_FCS_20210927_1128.txt
```

```
CRC_FCS_20210927_1133.txt
```

```
The script is executing.....
```

The script execution has completed

6. La secuencia de comandos 2 va a imprimir los datos en formato de tabla, como se muestra en el ejemplo siguiente.

Principalmente, va a enumerar las interfaces de nodo con errores CRC y FCS que no sean cero, junto con la diferencia en sus contadores CRC/FCS, durante el intervalo de tiempo especificado por el usuario. Con LLDP, la secuencia de comandos también va a determinar el dispositivo vecino conectado con interfaces determinadas y, lo más importante, va a indicar qué nodo/interfaz es la fuente de errores desde el punto de vista del fabric y qué interfaces de nodo están viendo simplemente CRC debido a Stomp.

Desde el punto de vista de la resolución de problemas de FCS, el resaltado en "Rojo" y marcado como "Local" es en el que se debe centrar la resolución de problemas.

Esto es probablemente la interfaz(s), donde los paquetes dañados/dañados ingresan al entramado desde y causan que los CRC se inunden en el entramado.

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| POD_ID | NODE_ID | NODE_NAME | NODE_ROLE | INTERFACE | 20210927_1128 | 20210927_1133 | 20210927_1128 |
20210927_1133 | NEIGHBOR | ERROR SOURCE |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 302 | bgl-aci06-t2-leaf2 | leaf | eth1/44 | 5002806823759 | 127841888 | 5002806823759 | 127841888 | No
LLDP /CDP neighbours found please check physically where this interface connects | Local |
| 1 | 101 | bgl-aci06-spine1 | spine | eth1/1 | 2981200154 | 132103050 | 0 | 0 |
System:bgl-aci06-t1-leaf1.cisco.com,Interface:Eth1/49 | Stomp |
| 1 | 101 | bgl-aci06-spine1 | spine | eth1/2 | 968286 | 0 | 0 | 0 |
| Historic |
| 1 | 201 | bgl-aci06-t1-leaf1 | leaf | eth1/1 | 12 | 0 | 0 | 0 |
| Historic |
| 1 | 201 | bgl-aci06-t1-leaf1 | leaf | eth1/51 | 4999243774529 | 0 | 0 | 0 |
| Historic |
| 1 | 201 | bgl-aci06-t1-leaf1 | leaf | eth1/52 | 5002807353809 | 127841212 | 0 | 0 |
System:bgl-aci06-t2-leaf2.cisco.com,Interface:Eth1/49 | Stomp |
| 1 | 202 | bgl-aci06-t1-leaf2 | leaf | eth1/51 | 968286 | 0 | 0 | 0 |
| Historic |
| 1 | 301 | bgl-aci06-t2-leaf1 | leaf | eth1/44 | 4999245287405 | 0 | 4999245287405 | 0 |
| Historic |
| 1 | 301 | bgl-aci06-t2-leaf1 | leaf | eth1/49 | 4999823953891 | 0 | 0 | 0 |
| Historic |
| 1 | 302 | bgl-aci06-t2-leaf2 | leaf | eth1/49 | 4999243774529 | 0 | 0 | 0 |
| Historic |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

7. Además, la secuencia de comandos proporcionará las siguientes opciones a los usuarios para ordenar y ver datos granulares, lo que recopilaron las secuencias de comandos 1 y 2.

El usuario puede elegir una opción entre el número 1-3 como entrada. Vea el ejemplo siguiente.

- 1.Sort the data further
- 2.View the granular data of an interface
- 3.Exit

Input the number:

En el siguiente ejemplo, vamos a la opción 2, que nos ayuda a ver los datos granulares de cualquier interfaz dada.

El script solicitará al usuario que introduzca el número de POD , ID de nodo e ID de interfaz correspondientes en la tabla que se muestra más arriba (paso 6).

Aquí en este ejemplo, estamos usando 1-302-eth1/44, donde el ID de POD es 1, el ID de nodo es 302 y el ID de interfaz eth1/44. Esta es la interfaz

donde el script notificó el FCS local, como se muestra en el paso 6.

Input the number:2

Enter an interface for which you need granular data(POD_ID-NODE_ID-INTERFACE Example:1-101-eth1/5): 1-302-eth1/44

You have CRC and FCS data in the below date range
1.2021-09-27

Enter the date for which you need granular data(any number from the above list range(1-1)):

En nuestro ejemplo, recopilamos los datos sólo durante unos minutos al día, por lo que vemos una única opción para fechada el 27 de septiembre.

Por lo tanto, nuestra aportación será "1".

Enter the date for which you need granular data(any number from the above list range(1-1)): 1

Time	CRC	FCS
11:28	5002806823759	5002806823759
11:33	5002934665647	5002934665647

Do you want to continue viewing the granular data(0/1), 1=yes, 0=no:0

Please select any number below to sort the data further or to view granular data of an interface

- 1.Sort the data further
- 2.View the granular data of an interface
- 3.Exit

Input the number:3

ABCD-M-G24X:downloads abcd\$